CTX-Configuration-Store User Guide

Contents

CTX-Configuration-Store User Guide 1
Contents 2
Versions 4
Document Revisions 4
Module Versions 4
Preface 5
About this Manual 5
Audience 5
Related Material 5
Abbreviations used in this Document 5
Requirements 6
Integration 7
Integration with Third-Party Systems 7
Integration with Existing Infrastructure 8
1 Configuration Management Overview 9
1.1 Using the Module 9
1.2 User Experience 9
1.2.1 Cortex-ConfigStore-Management-UI 9
2 Configuration Management Subtasks 16
2.1 Config-CGD-Get-DB-Server 16
2.1.1 Overview 16
2.1.2 Inputs 16
2.1.3 Outputs 16
2.2 Config-CQD-Query-DB 16
2.2.1 Overview 16
2.2.2 Inputs 17
2.2.3 Outputs 17
2.3 Config-CGP-Get-Parameters 17
2.3.1 Overview 17
2.3.2 Inputs 18
2.3.3 Outputs 18
2.4 Config-CEV-Encrypt-Value 18
2.4.1 Overview 18
2.4.2 Inputs 18
2.4.3 Outputs 19
2.5 Config-GCPQ-Generate-Config-Params-Queries 19

 © All Rights Reserved

2.5.1 Overview 19
2.5.2 Inputs 19
2.5.3 Outputs 20
2.6 Config-GPVDQ-Generate-Parameter-Value-Delete-Query 20
2.6.1 Overview 20
2.6.2 Inputs 20
2.6.3 Outputs 21
3 LivePortal UI Flows 22
3.1 Cortex-ConfigStore-Management-UI 22
3.1.1 Overview 22
3.1.2 States 22

 © All Rights Reserved

Versions

Document Revisions

The following revisions have been made to this document

Date	Revision	Notes
28/05/2019	1.0	First Release

Module Versions

The following revisions have been made to this document

Date	Revision	Notes
28/05/2019	1.0	Creation of: • Config-CGD-Get-DB-Server • Config-CQD-Query-DB • Config-CGP-Get-Parameters • Config-CEV-Encrypt-Value • Config-GCPQ-Generate-Config-Params-Queries • Config-GPVDQ-Generate-Parameter-Value-Delete-Query • Cortex-ConfigStore-Management-UI

 © All Rights Reserved

Preface

About this Manual

This document is a user guide for the CTX-Configuration-Store module.

Audience

The audience for this document is those wanting to understand how to use CTX-Configuration-Store module.

Related Material

Document
CTX-Configuration-Store – Deployment Plan
CTX-Configuration-Store.studiopkg

Abbreviations used in this Document

OCI Orchestration Communication Interface

DB Database

 © All Rights Reserved

Requirements

The CTX-Configuration-Store module requires the following:

- Minimum Cortex v6.4 installed on the Cortex Application Server
- SQL Cortex-Configuration-Store-Install database installed

If you have a previous Cortex Configuration Store (beta version before the GitHub release) please contact Cortex to obtain the migration document and scripts. These will allow you to migrate existing configuration data into this Configuration Store and continue using it as before.

 © All Rights Reserved

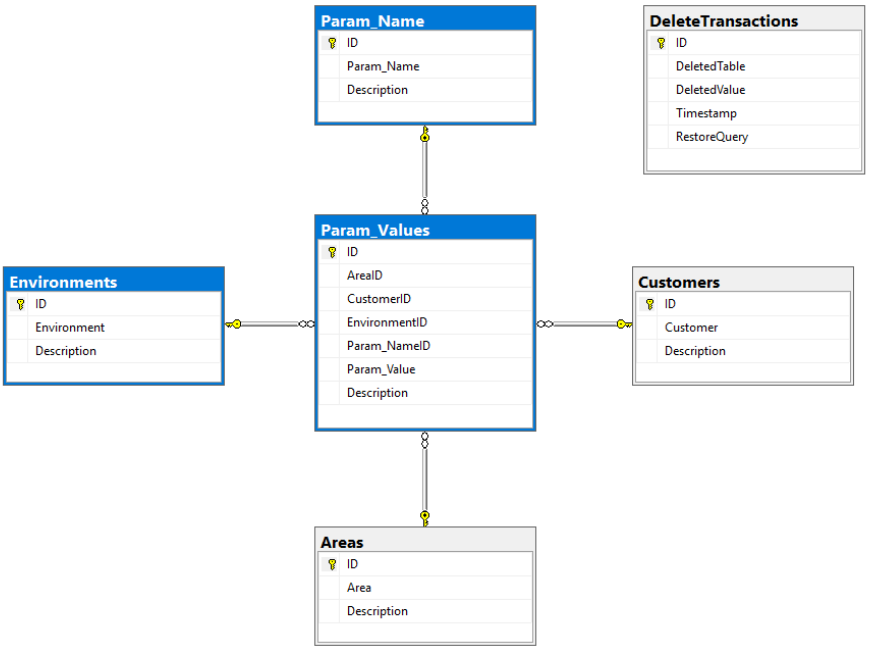
Integration

Integration with Third-Party Systems

For the flows and subtasks to work in the CTX-Configuration-Store module, the Cortex Config Store database and schema needs to exist on the server containing the Cortex databases. Instructions how to set this up are provided in the 'CTX-Configuration-Store – Deployment Plan'.

The tables involved in the Cortex Configuration Store schema are:

- Customers – Table containing the details on the customer
- Area – The area where the parameters are valid for. For example, 'Generic' could apply to all processes, 'SharePoint' would apply to interacting with SharePoint
- Environment –The environment the config parameters are valid for. Generally, one of the following:
    - Development
    - Pre-production
    - UAT
    - Production
- Param_Name – The name of the config parameters. For example, 'External-DB-Server' is a parameter where the name of a server where other databases are stored.
    - NOTE: The actual value (in the example, the server name) would not be stored here, instead it would be stored in 'Param_Values'.
- Param_Values – Stores parameter values and link to Customers, Areas and Environments
- DeleteTransations – Stores the deleted configuration items and the restore queries.



 © All Rights Reserved

Integration with Existing Infrastructure

None Required.

 © All Rights Reserved

1 Configuration Management Overview

'Configuration' is the setting of parameters that customise the behaviour of a Cortex solution. Examples include:

- Database connection strings
- IP Addresses
- Warning thresholds
- Other process specific

The Configuration Management module allows the storage of configuration data on a database and provides all the required subtasks to access and modify it. In addition, it provides a user interface implemented in Cortex LivePortal to manipulate the data.

1.1 Using the Module

Once the .studiopkg file has been imported, navigate to the 'Cortex-Configuration-Management-UI' flow and set the global variable 'G_DB_Name' to an appropriate value. This is the SQL Server Database Name which the Cortex Config store was installed as. By default this value is set to 'Cortex-Configuration-Store'. If the database was deployed with the default name, no update is required.

All functionality is handled using the Cortex Database Interface service user. This user must have the following permissions to the Cortex Config Store Database:

- db_datareader
- db_executor

The Configuration Management module has the below capabilities:

1. Configuration Management, which is achieved using LivePortal. See section 1.2
2. Using the Configuration, by using the subtask Config-CGP-Get-Parameters to get the configuration based on the Area, Environment and/or Costumer.

1.2 User Experience

1.2.1 Cortex-ConfigStore-Management-UI

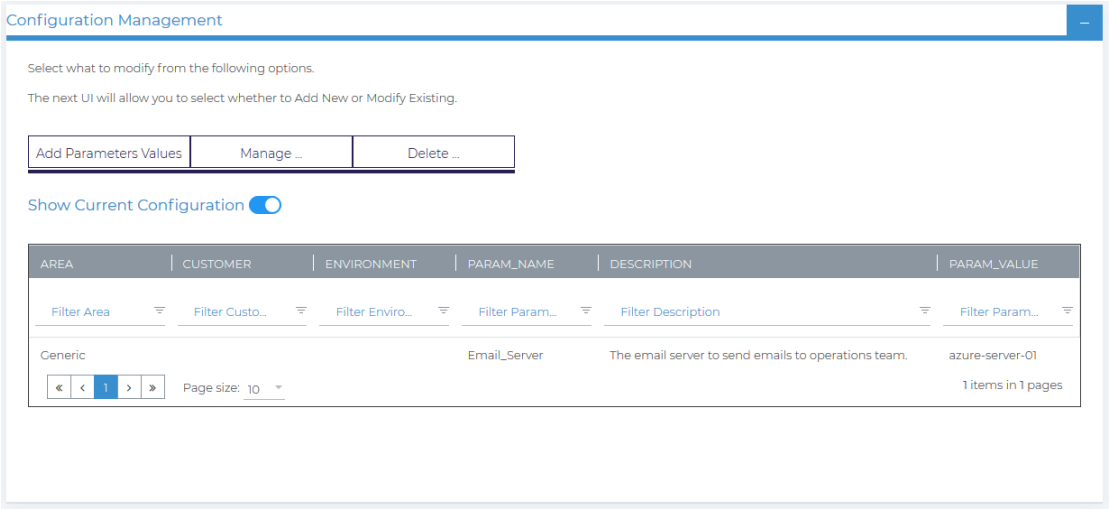
The 'Cortex-Configuration-Management-UI' is used by the user to create, edit and delete:

- Customers
- Areas
- Environments
- Parameters
- Parameters Values

1. When the flow starts, the user is presented with the **homepage**. This page allows the user to view the current configuration and select one of the following options:

 © All Rights Reserved

- Add Parameter Values
- Manage…
  - Areas
  - Customers
  - Environments
  - Parameters
  - Parameters Values
- Delete …
  - Areas (only visible for Advanced version)
  - Customers (only visible for Advanced version)
  - Environments
  - Parameters
  - Parameters Values



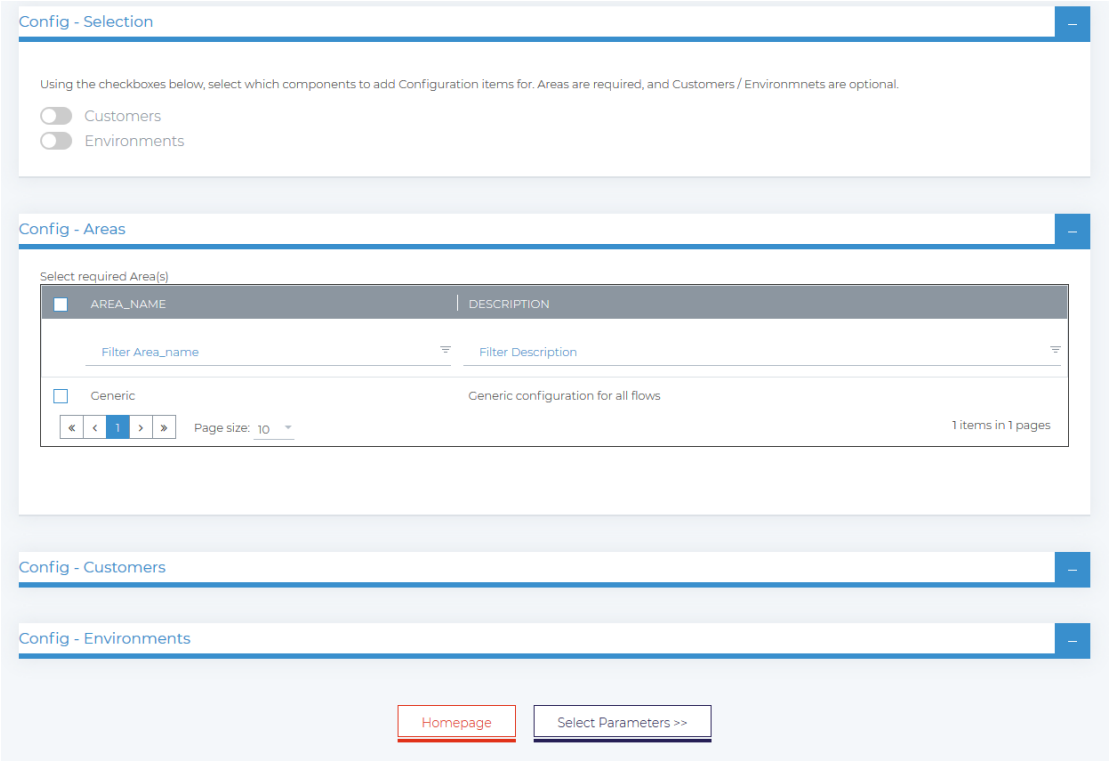
1.2.1.1 Add Parameters Values

To add Parameter values there must be a minimum of one Area and one Parameter already in the configuration database. Customers and Environment are optional as configuration can be created and mapped to only an area.

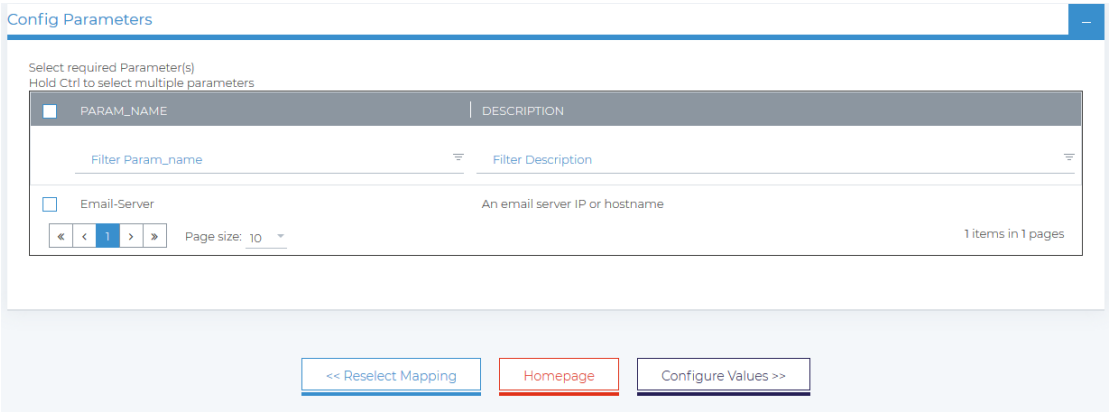
1. When the user selects to Add Parameter Values, a set of panels will be presented so that the user can select which mapping the configuration parameter(s) should have.
   a. The user must select one or more Areas
   b. Optionally the user can select one or more Environments and/or one or more Customers

Each combination of Area/Environment/Customer will be created to allow a bulk add of parameters

 © All Rights Reserved



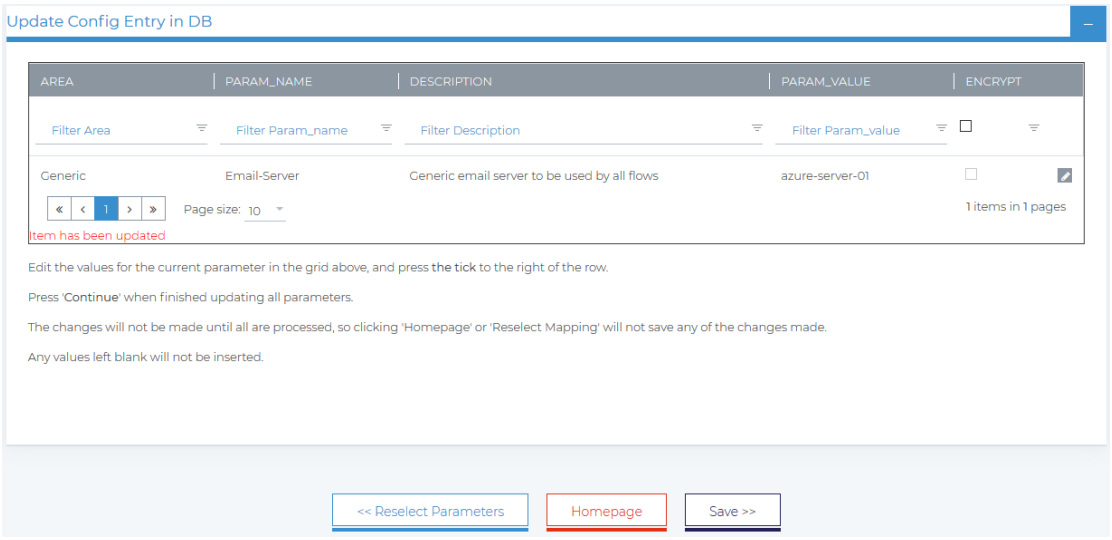
2. After selecting the required Areas, Customers and/or Environments, the user is presented with the Parameters selection screen.



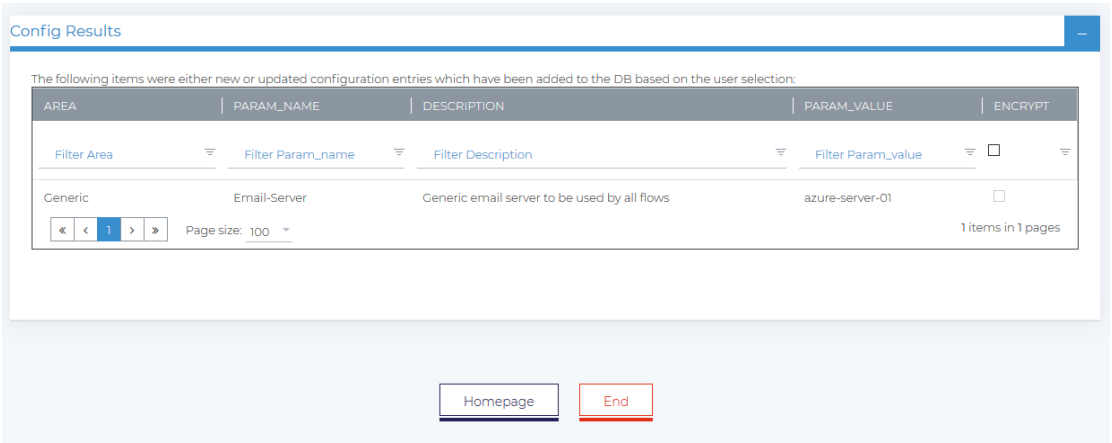
3. After selecting the required Areas, Customers, Environments and parameters to be configured, the user is presented with the edit screen.

- The screen will display all possible combinations of Area(s)/ Environment(s)/ Customer(s)/ Parameter(s). If one of the mappings already exists, it will be displayed and the user can either modify or leave it as is (no update will be done)
- It is possible to select specific values to be encrypted before being stored in the database

 © All Rights Reserved



4. After clicking Save >> the user will be presented will the results table

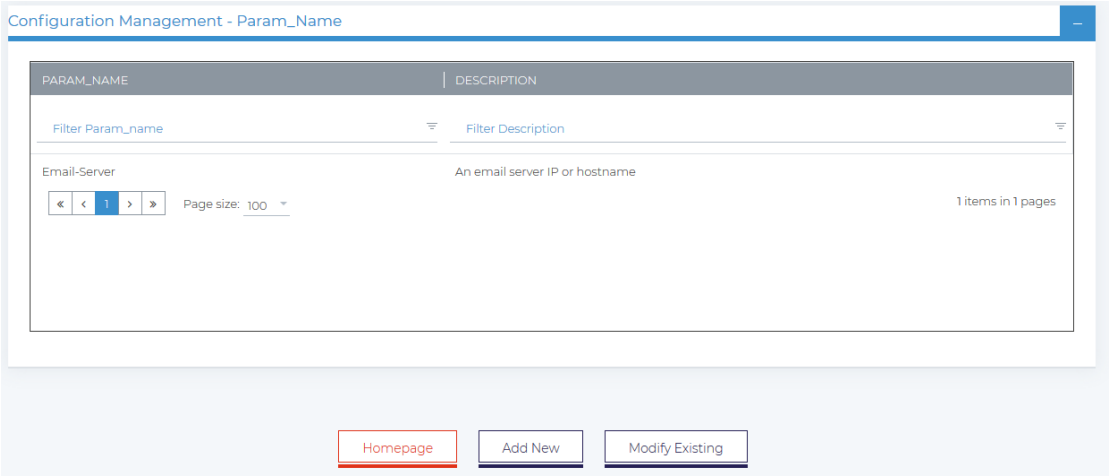


1.2.1.2 Manage Functionality (Areas, Environments, Customers, Parameters)

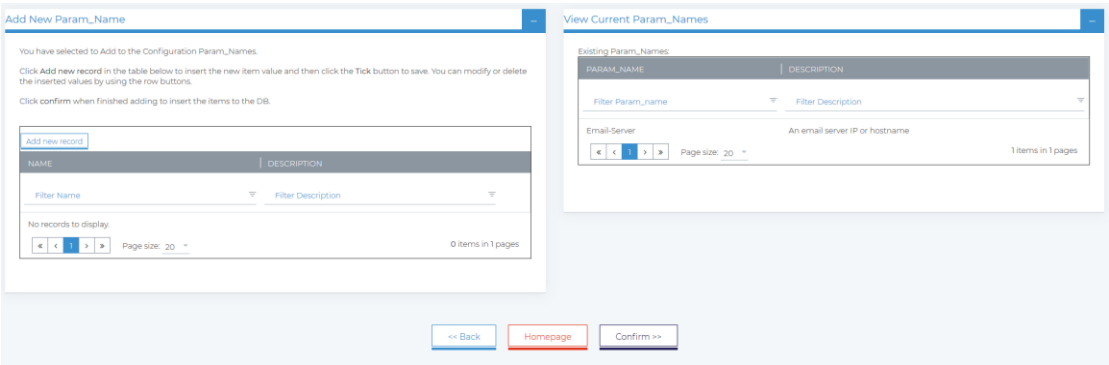
The manage user experience is the same when editing **Areas**, **Customers, Environments** or **Parameters**.

1. When the user selects to manage the configuration, a table is displayed with the current configuration. 3 options are displayed:
    a. **Homepage**, this will navigate back
    b. **Add New**, to add a new configuration item of that type
    c. **Modify Existing**, to modify the existing configuration items of that type

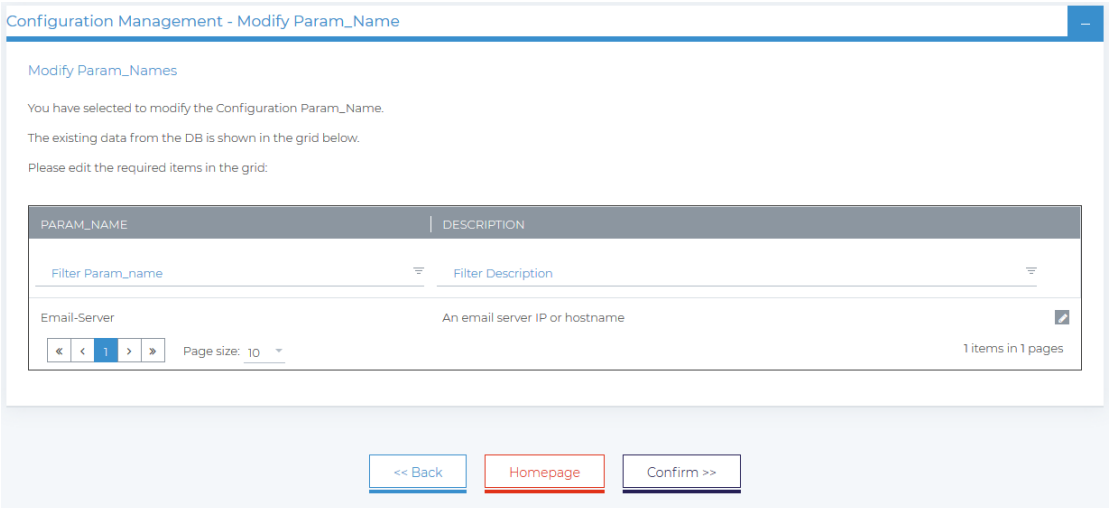
 © All Rights Reserved



2. If the user selects **Add New**, a new screen is presented with two panels. The one of the right side presents the existing configuration items of that type and the one on the left allows the user to add new ones



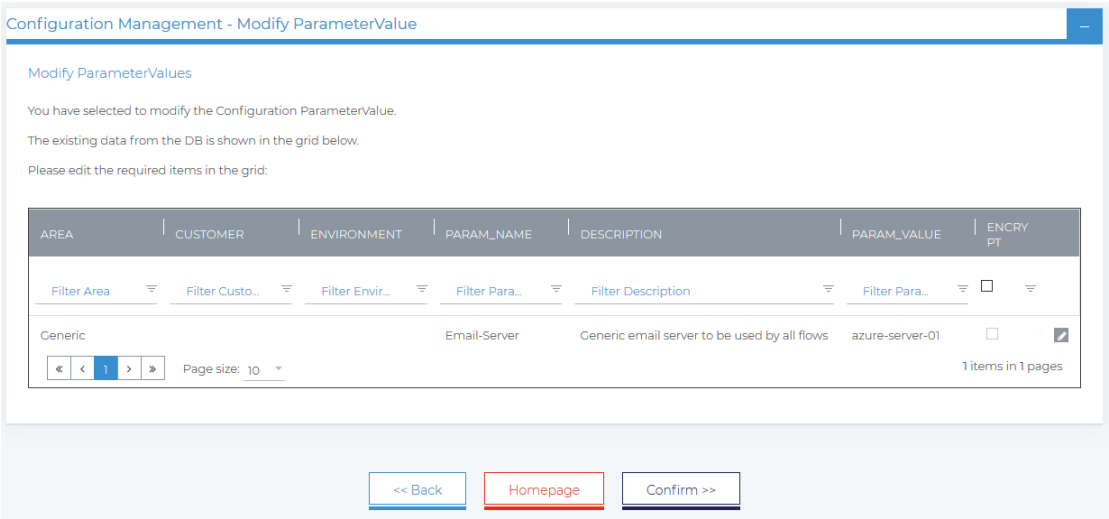
3. If the user selects **Modify Existing**, a new screen is presented the existing configuration items of that type and allows the user to edit them



 © All Rights Reserved

1.2.1.3 Manage Functionality (Parameter Values)

1. When the user selects to edit **Parameter Values** configuration, a table is displayed with the current parameter values and its linked configuration (Areas, Customers and Environments)



2. The user can edit the required values and commit the changes

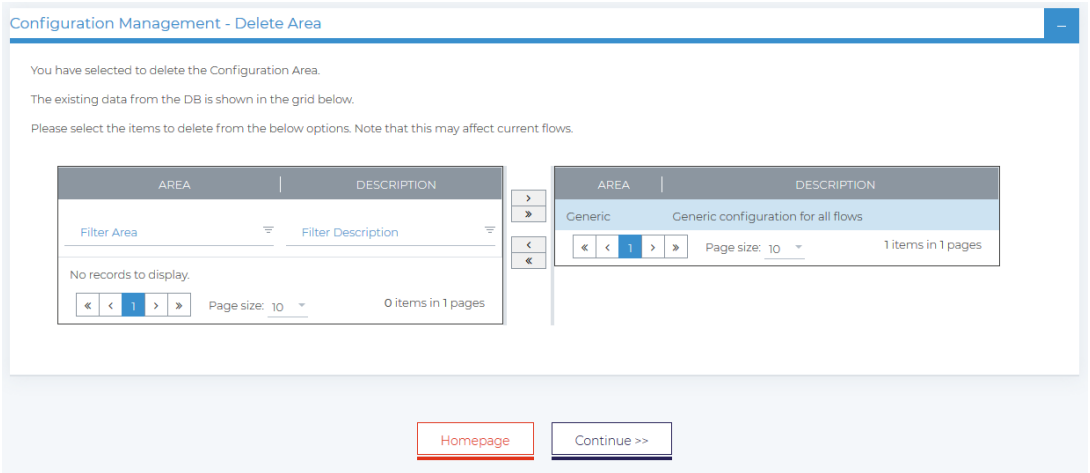
> Only rows with changes will be processed. Changes are allowed on the Description, Param_Value and Encrypt column.

1.2.1.4 Delete Functionality

The delete user experience is the same when deleting **Areas**, **Customers**, **Parameters** or **Parameter Values.**

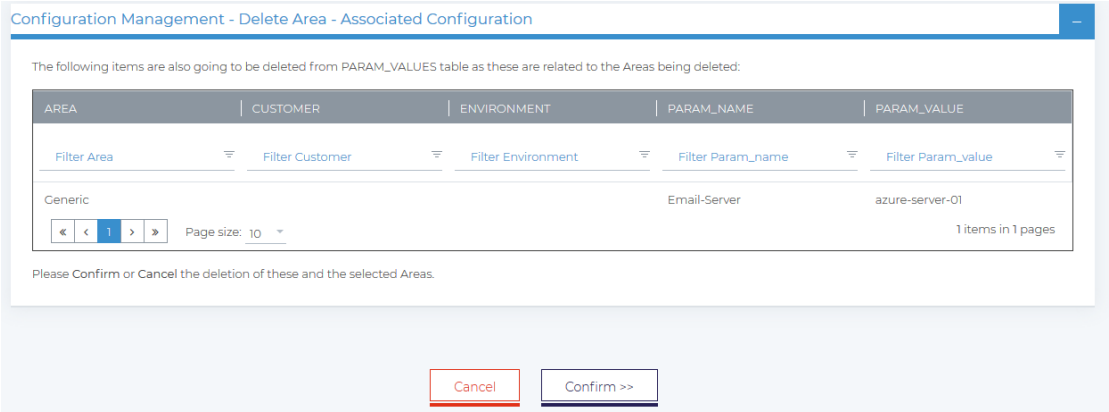
1. When the user selects to delete configuration, a double table is displayed with the current configuration of that item on the left side.

2. The items the user wishes to delete should be moved to the right side of the table.

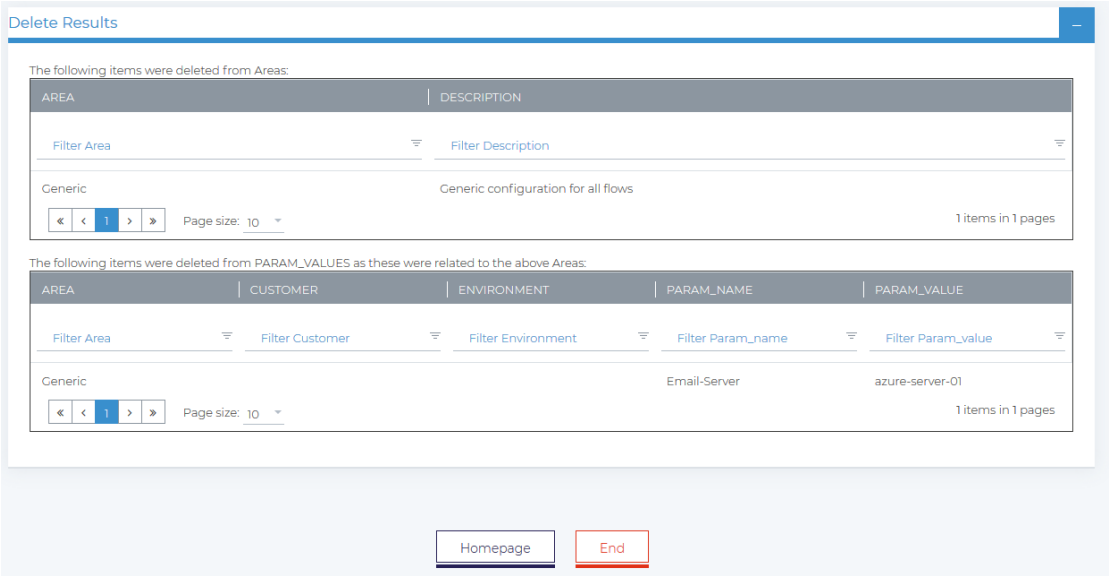


 © All Rights Reserved

3. When deleting an item which contains associated configuration, a new page displays all associated configuration which is also going to be deleted. The user can **Confirm >>** to delete both the item(s) selected before and these or **Cancel** the deletion.



4. After the deleting is performed a confirmation screen is displayed.



 © All Rights Reserved

2 Configuration Management Subtasks

2.1 Config-CGD-Get-DB-Server

2.1.1 Overview

The Config-Get-DB-Server subtask returns the server where the Reactor database is located.

2.1.2 Inputs

This subtask has no inputs.

2.1.3 Outputs

Output Variables	Type	Description
CGDS_o_SQL-Server	Text	The name of the server where the Reactor database is located.

2.2 Config-CQD-Query-DB

2.2.1 Overview

The Config-CQD-Query-DB subtask connects to a SQL database with the connection string specified and executes the SQL query specified.

Exceptions will be raised if:

- The connection string is not supplied
- The connection string is not valid
- The query is not supplied
- The query is not valid

 © All Rights Reserved

2.2.2 Inputs

Input Variables	Type	Description
CQD_i_Connection-String	Text	The connection string to the database. REQUIRED. Example: server=localhost;database=Reactor;trusted_connection=true
CQD_i_Query	Text	The SQL query to run on the database. REQUIRED. Example: SELECT * FROM CFG_Globals
CQD_i_results-as-list-boolean	Text	A True or False text value to control if the output should be given as a list or a table. Default value is False Example: False

2.2.3 Outputs

Output Variables	Type	Description
CQD_o_Results	Table	A table containing the result from the SQL query. Only populated if the input CQD_i_results-as-list-boolean is set to False.
CQD_o_Results-As-List	List	A list containing the result from the SQL query Only populated if the input CQD_i_results-as-list-boolean is set to True.

2.3 Config-CGP-Get-Parameters

2.3.1 Overview

The Config-Get-Parameters returns all parameters assigned to the Area/Customer/Environment combination specified.

Exceptions will be raised if:

- An Area has not been supplied

 © All Rights Reserved

2.3.2 Inputs

Input Variables	Type	Description
CFP_i_Area	Text	The Area to return the parameters for. REQUIRED Example: Azure
CFP_i_Customer	Text	The Customer to return the parameters for.
CFP_i_Environment	Text	The Environment to return to parameters for.
CFP_i_SQL-Server	Text	The name of the server where the Cortex Config Store database. If not supplied, 'localhost' will be used. Example: localhost
CFP_i_DB-Name	Text	The name of the Cortex Config Store database. If not supplied, 'Cortex-ConfigStore' will be used. Example: Config-Database

2.3.3 Outputs

Output Variables	Type	Description
CFP_o_Retrieved-Config	Structure	Config parameters assigned to the Area/Customer/Environment specified

2.4 Config-CEV-Encrypt-Value

2.4.1 Overview

Encrypts a value using the Cortex Flow API.

2.4.2 Inputs

Input Variables	Type	Description
EV_i_Flow-API-Config-Path	Text	The path where the Cortex Flow API is installed. Default value is "C:\Program Files (x86)\Cortex\Cortex Flow Interface Service\"

 © All Rights Reserved

		Example: Azure
EV_i_Value-To-Encrypt	Text	the value to be encrypted. REQUIRED. Example: Cortex

2.4.3 Outputs

Output Variables	Type	Description
EV_o_Encrypted-Value	Text	The encrypted value.

2.5 Config-GCPQ-Generate-Config-Params-Queries

2.5.1 Overview

This subtask is used as part of the configuration data generation. It is run for both environments and customers.

If they are required, the subtask will add to the SELECT query, the INSERT query headers, the JOIN query and the WHERE Clause for the Select query.

2.5.2 Inputs

Input Variables	Type	Description
GCPQ_i_Link	Text	Text value containing either Customer or Environment. REQUIRED Example: Customer
GCPQ_i_Insert-Query-Header	Text	Text value containing the start of the insert query. REQUIRED Example: INSERT INTO Param_Values (AreaID, ParamID
GCPQ_i_Select-Clauses	Text	Text value containing the start of the select query. REQUIRED Example: SELECT Area
GCPQ_i_Selected-Values	List	List containing the values selected for the specific link. REQUIRED Example: [“Dev”,”Prod”]
GCPQ_i_Read-Only-Columns	List	List containing the read only columns for the grid being displayed to the

 © All Rights Reserved

		user during the addition of new configuration values. REQUIRED Example: ["Area", "PARAM_NAME"]

2.5.3 Outputs

Output Variables	Type	Description
GCPQ_o_INSERT-Query-Header	Text	Text value containing the updated insert query.
GCPQ_o_SELECT-Clauses	Text	Text value containing the updated insert select query.
GCPQ_o_Selection-CSV	Text	Text value containing where clause for the specific link.
GCPQ_o_Read-Only-Columns	List	List containing the updated read only columns for the grid being displayed to the user during the addition of new configuration values.

2.6 Config-GPVDQ-Generate-Parameter-Value-Delete-Query

2.6.1 Overview

This subtask is used as part of the configuration data generation. It is used to generate the delete queries for parameter values.

2.6.2 Inputs

Input Variables	Type	Description
GPVDQ_i_Items-To-Delete	Table	Table of parameter values selected to be deleted. Table requires columns: Id, Area, Customer, Environment, ParamName, Description, ParamValue REQUIRED

 © All Rights Reserved

2.6.3 Outputs

Output Variables	Type	Description
GPVDQ_o_Delete-Query	Text	Text value containing the delete query.
GPVDQ_o_Delete-Transactions-Query	Text	Text value containing the rollback query to be inserted in the transactions table.

 © All Rights Reserved

3 LivePortal UI Flows

3.1 Cortex-ConfigStore-Management-UI

3.1.1 Overview

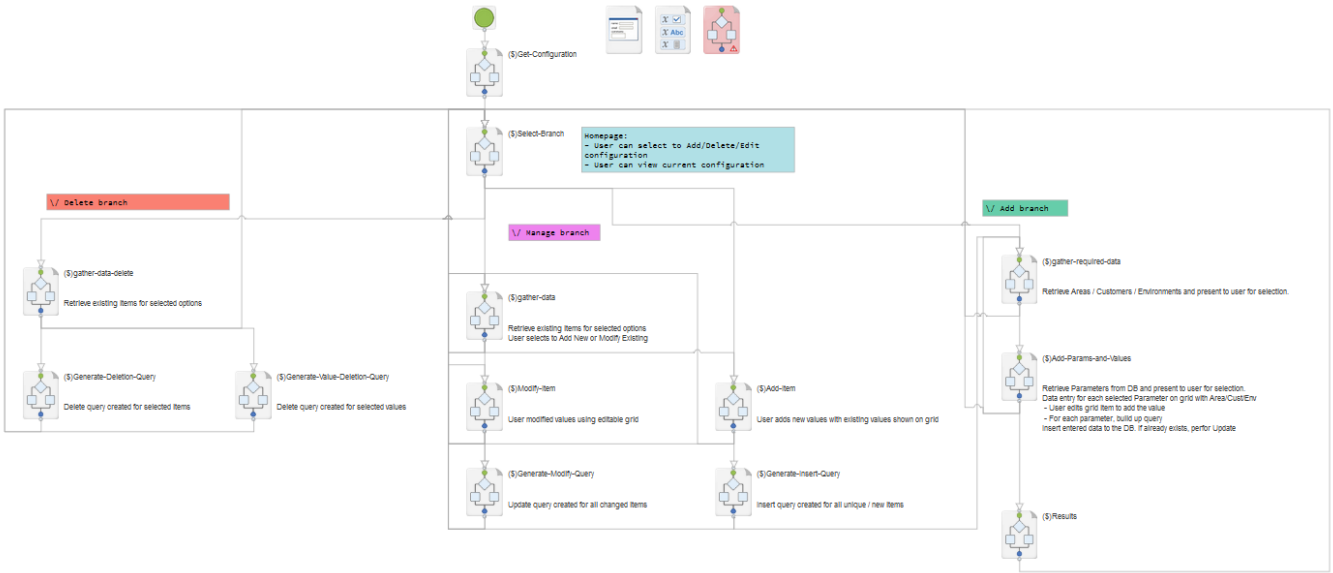
The 'Config-Manage-Configuration' flow is used by the user to create, edit and delete:

- Customers
- Areas
- Environments
- Config_Parameters
- Config_Values

The delete option is used to delete items from the Cortex Config Store. Rows from the following tables can be deleted:

- Customers
- Areas
- Environments
- Param_Name
- Param_Values

3.1.2 States



- Generic Branch
    - Get-Configuration

      Retrieves the database connection string. Uses the same SQL server as the Cortex reactor database and the database name specified in g_db-name.

    - Select-Branch

 © All Rights Reserved

The user will be displayed with a grid with the current configuration and a menu bar with the available options.

- Delete Branch
    - Gather-Data-Delete

    The user will be displayed a double grid with all items in the left grid. The user will have to move all items to delete to the right grid and confirm the update.

    - Generate-Deletion-Query

    A query is generated and executed to delete either Areas, Customers, Environments or Configuration Parameters. Once this is done, the user will be displayed a table containing the data which is now deleted in the database.

    - Generate-Value-Deletion-Query

    A query is generated and executed to delete either Parameter Values. Once this is done, the user will be displayed a table containing the data which is now deleted in the database

- Manage Branch
    - Gather-Data

    The user will be displayed all information in the table they selected. They will be able to either select the row(s) they would like to update (where they will be taken to the 'Modify Item' state) or add a new row (where they will be taken to the 'Add Item' state).

    Note: If the table selected is blank, the user will be instructed they are only able to navigate to the 'Add Item' state

    - Modify-Item

    The user will be presented a table containing all rows selected in the 'Gather Data' state.

    - Generate-Modify-Query

    A query is generated and executed to update all rows in the database the user has made modifications to. Once this is done, the user will be displayed a table containing the data from which have been updated in the database.

    - Add-Item

    The user will be presented a textbox where they are able to input a new item. The user will have to click 'Add', the item will be then displayed in a listbox to the user to the right of the textbox. Once the user has added all the required items to the listbox, the user will have to click 'Confirm' to add all items in the listbox to the database.

    - Generate-Insert-Query

 © All Rights Reserved

A query is generated and executed to insert all rows in the user has created in the 'Add Item' state. Once this is done, the user will be displayed a table containing the data from which have been inserted in the database

- Add Branch
    - Gather-Required-Data

    Returns all Customers, Areas and Environments in the database and the user will select which Customer, Area and Environment they are going to assign parameters to.

    - Add-Parameters-and-Values

    The user selects which parameters to link to the mappings selected in the previous state. Any parameters that are already linked to these will be shown with the current values. The user can either leave the configuration untouched or perform modifications.

    - Results

    The user is presented a results screen showing the updated/new data.

 © All Rights Reserved